# Homework 6

*Kitu Komya (404491375)*

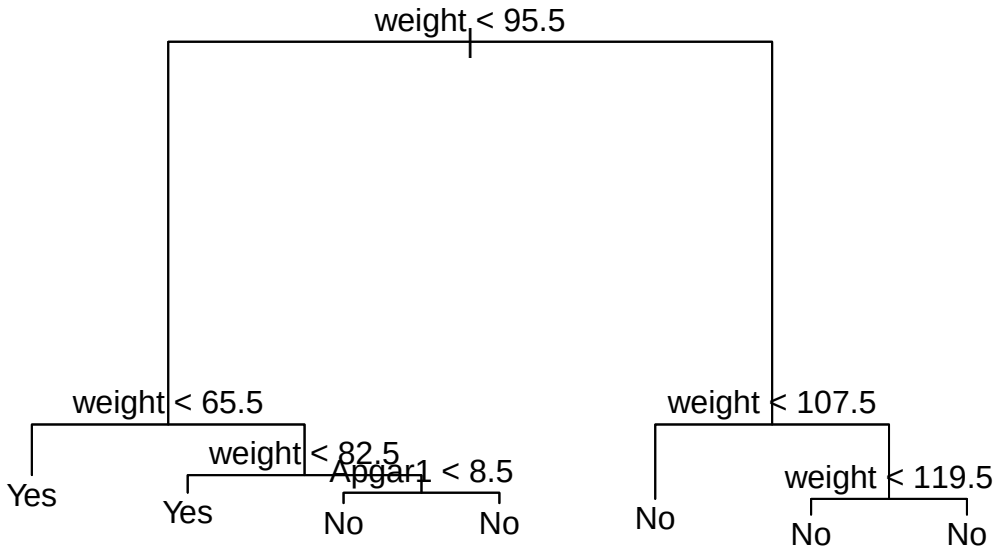
*June 1, 2017*

**1a**

```
library(tree)
better <- read.csv(file = "better2000births.csv")

# subset data
set.seed(9876)
training <- better[sample(nrow(better), 1000), ]
testing <- better[sample(nrow(better), 1000), ]

# make factor
training$Premie <- as.factor(training$Premie)
training$Gender <- as.factor(training$Gender)
training$Marital <- as.factor(training$Marital)
training$Racemom <- as.factor(training$Racemom)
training$Racedad <- as.factor(training$Racedad)
training$Hispmom <- as.factor(training$Hispmom)
training$Habit <- as.factor(training$Habit)
training$MomPriorCond <- as.factor(training$MomPriorCond)
training$BirthDef <- as.factor(training$BirthDef)
training$DelivComp <- as.factor(training$DelivComp)
training$BirthComp <- as.factor(training$BirthComp)

better$Gender <- as.factor(better$Gender)
better$Marital <- as.factor(better$Marital)
better$Racemom <- as.factor(better$Racemom)
better$Racedad <- as.factor(better$Racedad)
better$Hispmom <- as.factor(better$Hispmom)
better$Habit <- as.factor(better$Habit)
better$MomPriorCond <- as.factor(better$MomPriorCond)
better$BirthDef <- as.factor(better$BirthDef)
better$DelivComp <- as.factor(better$DelivComp)
better$BirthComp <- as.factor(better$BirthComp)

# make tree
tree <- tree(Premie~., data = training)
summary(tree)
```

```
##
## Classification tree:
## tree(formula = Premie ~ ., data = training)
## Variables actually used in tree construction:
## [1] "weight" "Apgar1"
## Number of terminal nodes:  7
## Residual mean deviance:  0.286 = 283.9 / 993
## Misclassification error rate: 0.056 = 56 / 1000
```

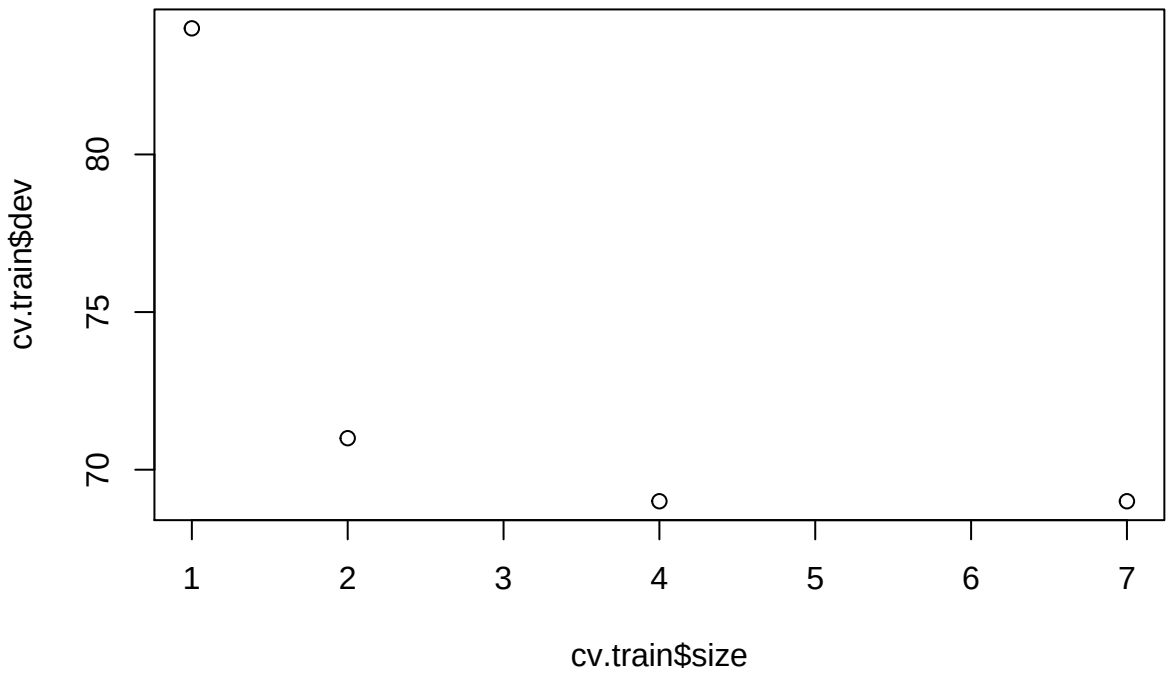
```
plot(tree)
text(tree, pretty = 0)
```



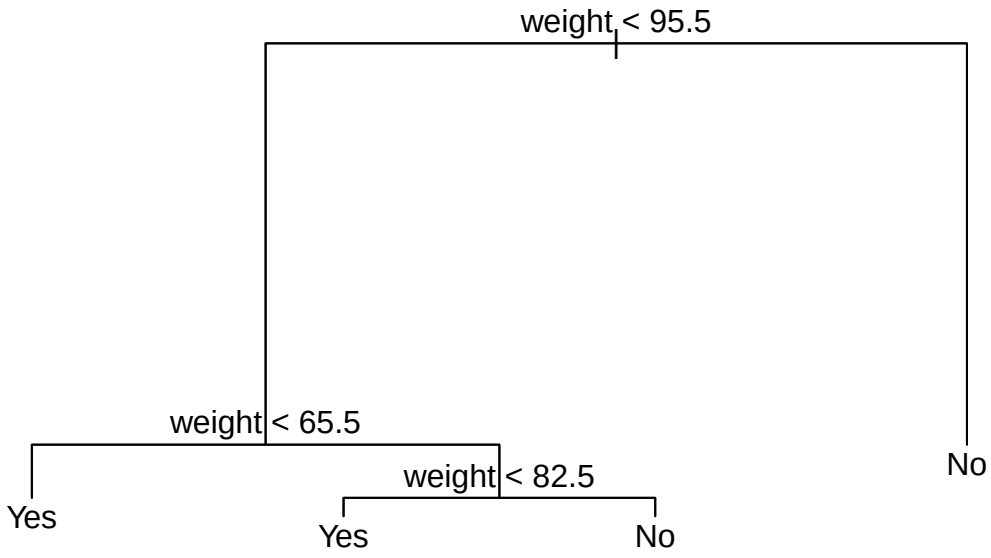

The testing misclassification error rate is 0.05611.

**1b**

```
cv.train <- cv.tree(tree, FUN = prune.misclass)
plot(cv.train$dev~cv.train$size) # we see 4 is the best size
```




```
pruned_fit <- prune.misclass(tree, best = 4)
plot(pruned_fit)
text(pruned_fit, pretty = TRUE)
```




```
summary(pruned_fit)
```

```
##
## Classification tree:
## snip.tree(tree = tree, nodes = c(3L, 11L))
## Variables actually used in tree construction:
## [1] "weight"
## Number of terminal nodes:  4
## Residual mean deviance:  0.3439 = 342.5 / 996
## Misclassification error rate: 0.056 = 56 / 1000
```

I have pruned my tree at terminal nodes 4 since size 4 is the best as seen through the plot.

**1c** We see the misclassification rate is the same from the pruned tree. The only variable that affects premature births is weight, so smoking is not a potential cause.

**1d** The testing misclassification rate is the same in the pruned and unpruned tree, which is 0.05611. So yes, we did do better, since our rate is 5.611% only!

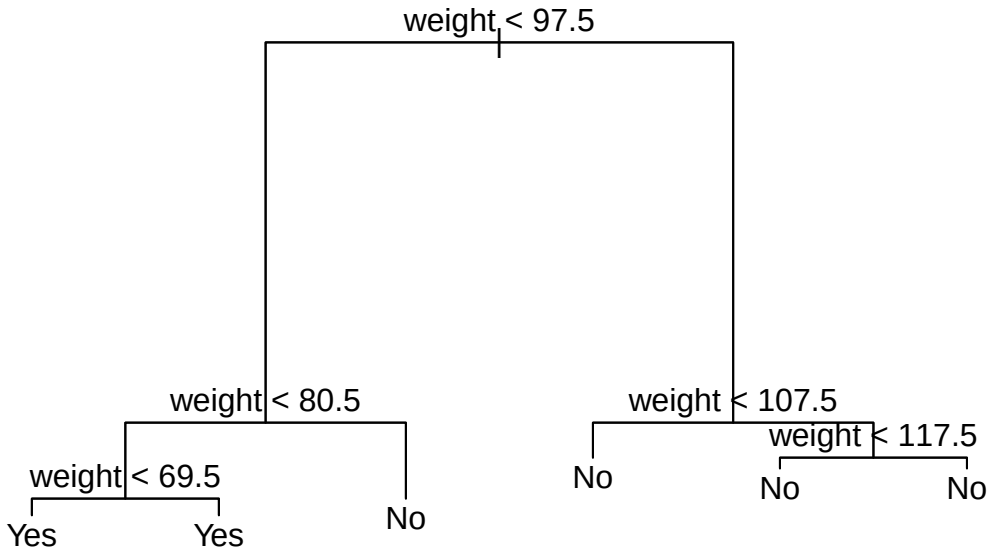
**2a**

```
tree2 <- tree(Premie~., data = better)
summary(tree2)
```

```
##
## Classification tree:
## tree(formula = Premie ~ ., data = better)
## Variables actually used in tree construction:
## [1] "weight"
```

```
## Number of terminal nodes:  6
## Residual mean deviance:  0.3132 = 624 / 1992
## Misclassification error rate: 0.05556 = 111 / 1998
```

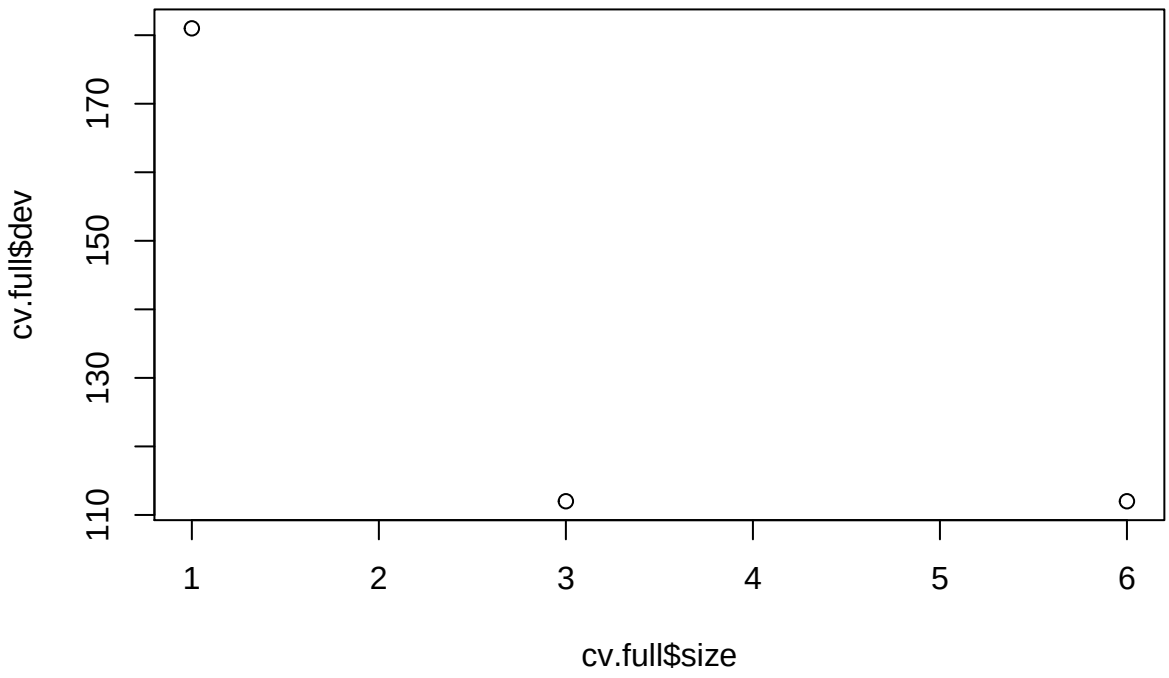
```
plot(tree2)
text(tree2, pretty = 0)
```



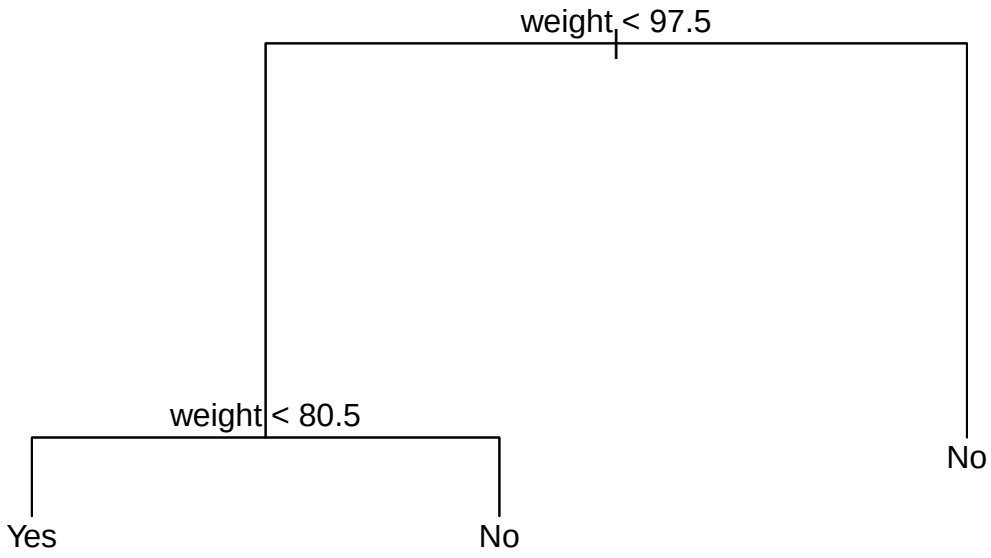

On the full data, the residual mean deviance is 0.3132, and the misclassification error rate is 0.05556.

**2b**

```
cv.full <- cv.tree(tree2, FUN = prune.misclass)
plot(cv.full$dev~cv.full$size) # we see 3 is the best size
```




```
pruned.fit <- prune.misclass(tree2, best = 3)
plot(pruned.fit)
text(pruned.fit, pretty = TRUE)
```




```
summary(pruned.fit)
```

```
##
## Classification tree:
## snip.tree(tree = tree2, nodes = 3:4)
## Variables actually used in tree construction:
## [1] "weight"
## Number of terminal nodes:  3
## Residual mean deviance:  0.3484 = 695.1 / 1995
## Misclassification error rate: 0.05556 = 111 / 1998
```

The tree has been pruned to size 3, since that's the best size as suggested by the plot.

**2c**

We see the misclassification rate is the same from the pruned tree. The only variable that affects premature births is weight, so number of visits is not an important feature.